Real-time command and control of nanosatellites

Alexandre Helmer*, Félix Petit*, Fabien Apper†, Clément Chalumeau† and Nicolas Humeau†
*Student of Space Eng., Institut Supérieur de l'Aéronautique et de l'Espace, France
Email: {felix.petit, alexandre.helmer}@student.isae-supaero.fr †System Engineer, Institut Supérieur de l'Aéronautique et de l'Espace, France
Email: {fabien.apper, clement.chalumeau, nicolas.humeau}@isae-supaero.fr

(Final paper)

***Abstract*—Being able to operate a spacecraft from the Earth is a crucial point in any space mission, in particular in the case of nanosatellites, which are usually visible a few minutes per day from the ground station. This low visibility time is problematic, not only because you have to listen to the satellite at a precise moment each day, but also because it makes it difficult to make big changes and corrections from the ground.**

Our aim is to use a satellite phone constellation (Iridium in particular) to establish bidirectional communications with the Earth. For that, we needed to study the feasibility of a command-control link through satellite constellations and to adapt the existing systems that were designed for the S band, which includes writing additional parts of code to the control center, configuring an Iridium modem and writing a driver for the modem as part of the flight software.

As a result of our study, it should be possible to use a basic command and control through Iridium with a significant improvement in the visibility of the satellite from the Earth.

I. Context

A. Direct telecommunication with ground stations

Almost all satellites currently communicate with the ground with a direct link to the ground station antenna. This allows for a relatively short delays (about 1 ms) between an emission and the reception, essential because of the protocols requiring acknowledgment of each received packet, and a well defined coverage time. But to increase coverage time, many ground antennas in various places on earth are required. Yet, some areas cannot be covered and the antennas required to communicate with the ground are very constraining on nanosatellites energy-wise.

B. Satellite phone constellations

A new way to communicate is to use satellite phone constellations. The advantage is double: it firstly provides a better coverage, and secondly, as the satellite phone network is already in place and functional, using it should be reliable and cost effective. For these reasons, the public interest for this solution is growing. Our work will be to study the idea of a command and control system using satellite phone constellations, using the Eye-Sat control center and communication boards.

C. Choice of the constellation

Different kind of constellations exist, with their advantages and drawbacks: Geostationary Orbit (GEO) or in Low Earth Orbit (LEO), different coverage zones, different signal bands and data rates. (see Table I)

D. Problem statement

Our goal is to develop a way to command and control nanosatellites. We will study the feasibility of the use of a constellation such as Iridium or Globalstar for real-time command-control, and then focus on Iridium to design a command-control system, study the network aspects and link management, and study the mechanisms of on board time management in an asynchronous architecture.

II. Preliminary Theoretical study

In order to design a command-control system with Iridium, the following points have to be addressed :

TABLE I
COMPARISON OF DIFFERENT SATELLITE PHONE CONSTELLATIONS

Constellation	Orbit	Coverage	Band	Satellites (Act./Tot.)
Iridium	780 km polar	Total	L	39 / 77
Globalstar	1414 km 52°	Total	L	40 / 48
Inmarsat	GEO	$-82° S$ to $+82° N$	Ka	3/4
Thuraya	GEO	Africa, Asia, Europe, Australia	L	2/3
Orbcomm	750 km 45°	Variable	L (data)	12 / 18

- The impact of the attitude of the satellite on the signal [4]
- The impact of the Doppler effect on the signal
- The delay between the emission of a packet and its reception [6]
- The impact of the orbit of the satellite on the visibility by the constellation

These parameters have an influence on the quality of the signal once established, and on the visibility of our satellite by the constellation.

A. Signal Quality

1) Attitude control: Taking into account that the patch antennas have a beam width of 150°, if we wish to maintain a communication of about 5 minutes (similar to a satellite-ground direct link), the satellite's angular speed should not be over $0.5 \deg / s$. This requires attitude control, either active, or as described in [4], passive and function of the initial angular motion of the satellite.

However, once in the visibility cone, the satellite is in the exact same situation as in the direct link case. The constraints considered here being the same with Iridium or direct link and thus non determining, **we choose to not take them into account in our study.**

2) Impact of the Doppler effect: The Doppler effect is already present in direct communications. It comes from the relative speed of the satellite compared to that of the ground station, and the frequency shift can cause a gain loss on the receiving antenna. If the relative speed is relatively small compared to the bandwidth of the signal, the Doppler shift can be neglected [5]. With Iridium communication, the relative speed between the nanosatellite and the receiving Iridium satellite must be determined to check if the Doppler shift can be neglected.

We consider the worst case of Doppler shift, in which our satellite and an Iridium satellite orbit in the same plane and come from opposite directions. The Iridium satellite goes at 7.46 km/s and orbits at 780 km, the mission satellite flies at 7.66 km/s at 400 km. The frequency variation due to the Doppler effect is then about 80.6 kHz, which can be easily neglected on communications around 2.4 GHz.

As a consequence, the Doppler shift has a negligible impact on our systems.

3) Delay between the emission and the reception of a packet: With direct communications, the signal travels in a straight line between the satellite and the ground antenna, so there is a delay of some milliseconds between emission and reception, allowing protocols to have acknowledgment packets. In the case of a satellite phone constellation, the signal has to travel from the nanosatellite to the nearest satellite of the constellation, which will the send the data to other satellites, before the signal is sent to the ground station. This causes possible delays of **less than 20 seconds** according to Iridium documentation of SBD, while article [6] says a few seconds are needed for the message to be sent and received, and a few more for it to be processed.

B. Visibility determination

Remains to determine the influence of the orbit on the visibility. Considering circular orbits, only 2 parameters are determining: the altitude (directly related to the semi-major axis) and inclination. To control these parameters in order to get the "best" coverage, we define the quality of a configuration through the average session length (a session is an uninterrupted visibility period) and the percentage of visibility time during 24 hours (interpreted as the probability to be visible at a given time).

For the different calculations and simulations, we decided to use the Celestlab Library provided by the Centre National d'Études Spatiales (CNES) for Scilab.

1) Our work on Scilab: Scilab and the Celestlab library allow us to calculate and simulate the

The Two Line Element (TLE) is a standardized way to describe the position and the orbital parameters of an object orbiting the Earth at a given time. With the TLE of an object, one could in theory reconstruct its trajectory.
The data is encoded in ASCII on 3 lines, the first one being the name of the identified object. Then come its time-related parameters (date of the measure, derivative of the mean motion, date of launch) and identification attributes (name, id, classification). The last line contains the orbital parameters.
Example: TLE of the ISS

```
ISS (ZARYA)
1 25544U 98067A   08264.51782528 -.00002182  00000-0 -11606-4 0  2927
2 25544  51.6416 247.4627 0006703 130.5360 325.0288 15.72125391563537
```

Fig. 1. TLE - Two Lines Elements

visibility and coverage of a given satellite by a telecommunication constellation.

To calculate the coverage by a constellation at a certain altitude, we first imported the TLE of that constellation (see fig. 1) into Scilab from the North-American Aerospace Defense Command (NORAD) database [1].

Then we used Celestlab functions to help us generate the constellation at a given time, and take away the satellites which were not operational, from the data previously extracted.

We then wrote Scilab functions to determine if a satellite is seen by the constellation or not. From the data provided on Iridium website, we know that the Iridium antennas' half-power beam angle is wide enough to intercept the Earth. That defines a visibility cone by the tangent lines to the Earth. Outside that cone, the possibility of a communication is uncertain. We decided to not take the exterior of that cone into account.

Eventually, we also wrote scripts to run simulations over time, one satellite being on a predefined orbit, and to get the visibility parameters at any moment, like the number of visible phone satellites, the length of the current session and the Doppler effect.

2) Validation: In order to validate our tools, we decided to calculate ground coverages and compare them to those provided by Iridium or Globalstar websites. For this, we simply check for every latitude and longitude if the point is visible by any satellite, and obtain the ground coverage for a constellation.

Fig. 2. Calculated Iridium cover on the ground with all satellites

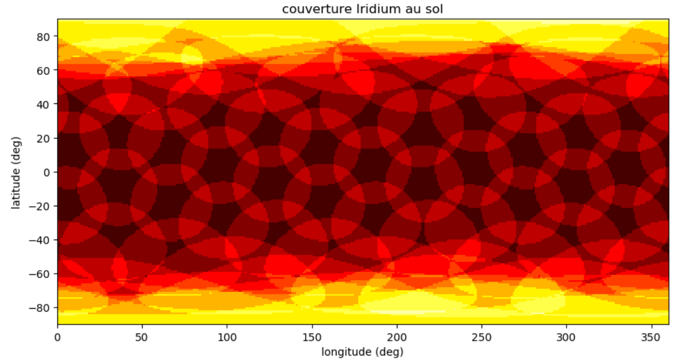
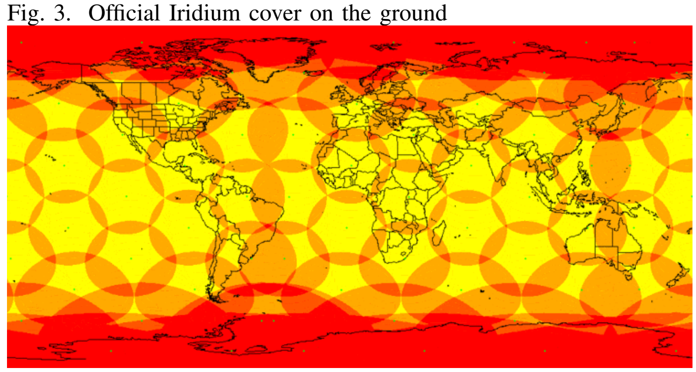


Fig. 3. Official Iridium cover on the ground



After comparing the ground coverage of a perfect 66-satellites constellation with the picture found on Iridium website (see fig. 2 and 3), we conclude to the validity of our coverage calculation.

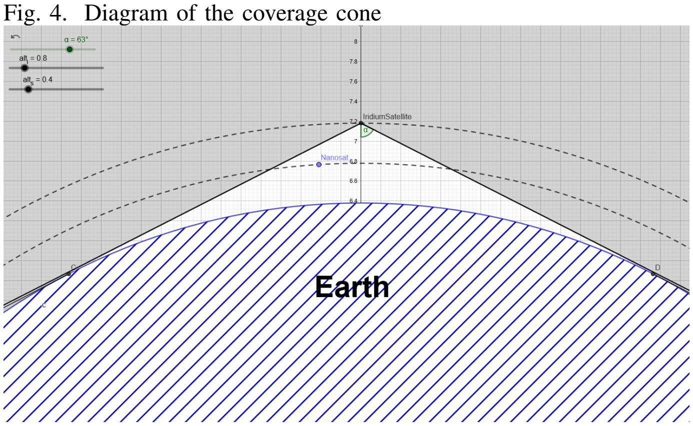
For later use, and for more interesting results, we then calculated real coverages (hence without disabled satellites) of Iridium, Iridium Next and Globalstar, on ground and at 400 km (See Appendix B).

3) EntrySat Validation: We also ran a simulation for EntrySat's orbit in order to calculate the probability of coverage at the reentry at 120 km. At this altitude, the satellite burns into the atmosphere and needs to transmits its data through Iridium, as the ground station will probably not be visible. We found a coverage probability of 94% with the theoretical Iridium constellation (66 satellites), and 90% with the real constellation (active satellites of both Iridium and Iridium Next) on June 22nd, 2018. The value of 94% is coherent with what was estimated by the EntrySat team.

4) Function of altitude and inclination: Once our scripts were tested, we could run simulations. For that, we chose an orbit (ISS for instance: 51.8° at an altitude of 400 km) and simulated the satellite's movement during 24h. From this simulation, we extracted information about the session's average length and the coverage time.

As seen in the coverage maps, the higher the satellite, the smaller the coverage. It can be understood from the following diagram (fig. 4):

Fig. 4. Diagram of the coverage cone



The inclination of the satellite will also impact the coverage, as most constellations do not have a homogeneous coverage on earth (for example, the Iridium constellation has a better coverage on the poles compared to the equator since its satellites follow polar orbits).

Consequently, we ran the simulations over different altitudes and inclinations for a given constellation.

The results (see Appendix C) confirm that the coverage diminishes with the altitude. The best inclination is the same as the one of the constellation. The visibility and the average session length seem to be highly correlated.

C. Costs

The Iridium communication costs are about $1 per kB. The Iridium transceiver costs $141. For a classic direct link, a UHF/VHF transceiver costs $8500 (space certified), and installing a ground station costs around $50 000 on ISISspace.

D. Conclusion

As a conclusion, we compare the Iridium link with the direct link on the following points:

- the session length
- the visibility percentage
- the maximum data rate
- the operating cost

(See Table II)

We do not take the signal quality into account, as the effects of the attitude and the Doppler shift are the same in both cases.

TABLE II
COMPARISON OF THE TWO METHODS FOR AN ISS ORBIT

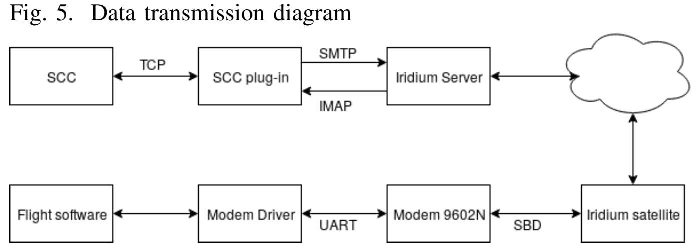
Criterion	Iridium	Direct UHF/VHF
Session length	114s	300 s
Visibility	24.7%	1.39%
Data rate	40 bps	up to 9600 bps [3]
Operating costs	$1/kB	human operator, costs may vary
Installation costs	$141	$8500 + 50000

III. PRACTICAL WORK

A. Presentation

To effectively test the communication with a nanosatellite using the Iridium constellation, we will implement the following data transmission chain (fig. 5).

Fig. 5. Data transmission diagram



The Simple Control Center (SCC) is a control center software developed by the CNES and ISAE-Supaero for the EyeSat mission. We also use EyeSat's flight software for our tests, so that we can focus on the Iridium parts: the modem, its driver, and a module to enable the SCC to communicate with Iridium.

B. The SCC plug-in

To send or receive a message to or from a Iridium transceiver identified by its International Mobile Equipment Identity (IMEI), we use emails. The subscription we use to send message with the Iridium constellation is linked to our email accounts. More practically, to send a message to the transceiver, one needs to write an email with the following format:

Destination: DATA@sbd.iridium.com
Subject: IMEI number of the transceiver
Attachment: message.sbd

where the IMEI references the transceiver, and message.sbd is a file with a sbd extension less than 270 bytes. We get a confirmation that our message has been queued for delivery on the Iridium satellites, along with its position in the queue.

When the transceiver sends a message, the Iridium servers send an email to the linked email accounts. Attached to the email is the message sent with an sbd extension. The content of the email gives the approximate position of the transceiver calculated with Doppler shift, the Mobile Originated Message Sequence Number (MOMSN), the Mobile Terminated Message Sequence Number (MTMSN) and the time of the transfer.

To generate the Telecommand (TC) packets and check the response from the On-Board Computer (OBC), we will use a control center software called SCC. This software generates TC packets from a database and sends them to a port with a TCP protocol. In the same way, the responses from the satellite are acquired through another port and analyzed by the software.

To ensure the communication via Iridium, we developed a bridge between the SCC and an email account. This plug-in connects to the ports specified in the SCC configuration, and monitors the inbox of one of the email accounts linked to the transceiver.

When sending a message from SCC, the TC packet is retrieved by the plug-in from the specified port and written in a sbd file. The plug-in then creates an email with the sbd file as attachment, the IMEI as subject and sends it to the Iridium servers. The plug-in checks the inbox for the confirmation email form Iridium to ensure no packets are lost.

When the responses are sent back, we receive an email from Iridium. The plug-in monitors the inbox, and when it receives an email from Iridium with an attachment, it extracts the sbd file from it, and sends only the data contained inside to the SCC.

We have tested the emission from SCC, as well as retrieving data from the email. The plug-in can be further developed to be fully integrated in the SCC so that one can choose to which transceiver the data is sent, and configure the plug-in from the software itself.

C. *The Iridium Modem*

To communicate with the Iridium constellation, we have an Iridium Transceiver called Iridium 9602. Its integration on a cubesat is made possible by its size (41*45*13 mm^3), its weight (30g) and its low power consumption. The communication with the Modem can be done with Universal Asynchronous Receiver Transmitter (UART), either using 3-wire or 9-wire. To test it on the Enclustra OBC, we chose the 3-wire UART, as it allows for sufficient control of the modem, and is easier to integrate. This transceiver uses the Short Burst Data (SBD) Iridium protocol.

The modem allows for different baud rates; we chose 115200 to ease the configuration on the OBC side.

Regarding the communication with Iridium, the Modem has two buffers, one for Mobile Originated (MO) messages, sent by the satellite, and another for the Mobile Terminated (MT), received by the satellite. The buffers are limited to a single message (340 bytes for MO and 270 for MT). To get information on the statuses of these buffers, we use the command "AT+SBDS", and wait for the response of the form "+SBDS: n_1, n_2, n_3, n_4", where n_1 and n_3 are the state of the MO, respectively MT, buffer, and n_2 and n_4 the MOMSN respectively MTMSN.

To write a message in the MO buffer, one can use two commands: "AT+SBDWT" which writes text (works only for ASCII characters encoded on 1 byte) or "AT+SBDWB=<length>" (where length is the number of bytes written), which writes a binary string. We will use the second command, as the TC packets and their responses are binary. The same way, we can either read text "AT+SBDRT" or binary "AT+SBDRB" from the MT buffer. The binary received will have a header of two bytes containing the length of data, and two bytes of checksum at the tail.

To send the message in the MO buffer to the constellation, or get MT messages from the constellation, we need to initiate a SBD (Short Burst Data) session. When issued an "AT+SBDI" command, the Modem will try to connect to an Iridium satellite, send his MO message, and receive any MT message that is queued in the constellation. If the Modem does not manage to connect to the constellation after about 20 seconds, it sends a response and stops its session.

D. The Iridium Driver

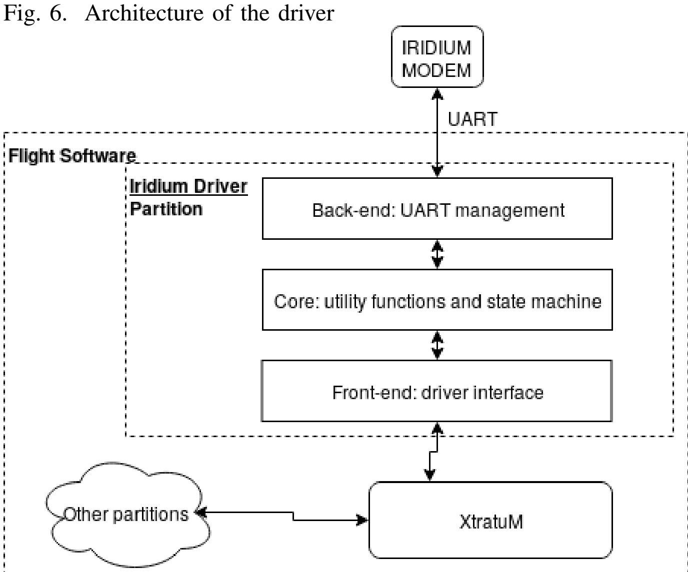
The driver is a program that runs on the OBC and makes the interface between the flight software and the Iridium Modem.

The driver is constituted by 3 layers:

- Back-end: ensures that the connection between the modem and the computer is done. Provides low-level functions such as ”write_serial” and ”read_serial”.
- Core: use the modem as it has been described before, to provide useful functions such as ”set_echo”, ”get_status”, ”write_message_buffer”, etc. The core uses mostly back-end functions.
- Front-end: uses core function to let the flight software send and read messages through the modem.

The driver is written in C and is synchronous. (see fig. 6)

Fig. 6. Architecture of the driver



During the implementation of the code, we ran through some difficulties due to the specific architecture.

1) Real-time management: Our driver should work in real time, and not be blocking. Which means that when the flight software calls a function to write a message, the driver should not wait until the emission is done. Instead, we add the message to a queue (which is instantaneous), and a background state machine manages the communications and the message queues for sending and receiving (see Appendix A).

2) Flight software implementation: As our driver will be part of the flight software, we need to understand its structure and the way it works when it goes about communicating. We firstly developed the core and back-end layers that we tested without the flight software, to check that the communication with the modem was correctly done. Then we used the front-end layer of an S-band driver, that we modified so it uses our core functions and the state machine. When the read and write functions from the front-end are called, the messages are added to queues which are processed by the core.

3) XtratuM time management and partitions: The main program that runs on the OBC, XtratuM [2], is a real-time supervisor for embedded systems. It gives our driver some time to execute, and stops it after. Which means that it is impossible to have functions in the driver that take more than 0.1 s to execute.

Problem: the ”AT+SBDI” command can potentially take a huge time (up to 20s) to finish. We can neither wait for its result, nor retrieve it from the UART buffer, too small. Thus, we decided to start the command, and regularly check its completion by sending ”AT” messages to the modem. If the modem does answer ”OK”, it means that the Iridium session has been established. Otherwise, we continue to check.

IV. Results

To conclude on the feasibility of using satellite phone constellations to communicate with nanosatellites, we firstly address the theoretical level, then focus on actual implementation.

Regarding the theoretical results, the table II shows that while the length of the sessions are in average slightly worse for the Iridium constellation than for a direct link, the percentage of visibility justifies the use of this method. Even at 400 km, the coverage is much better with Iridium or Globalstar than with the usual UHF/VHF. The biggest issue is the very low data rate available with Iridium, causing the total amount of data we can transmit to be less than a direct link. Hence, it’s an affordable solution for command and control, but is not suitable to retrieve huge mission data.

For what concerns the implementation, all the subsystems (the modem, the driver and the SCC plug-in) have been tested separately in both directions and were functional. We also tested the command-control chain by configuring the driver to answer to a PING request from the control center, which worked. However, we met a problem while testing the whole chain with the flight software, as it received our packet, but was unable to send back a reply, probably due either to a wrong configuration of the Space Packet Protocol, or to the involuntary disrespect of this norm by our program.

Our tests on Earth showed that we can expect to read a message every 8 seconds when covered by a satellite. This allows us to conclude positively on the use of Iridium to command the satellite and receive housekeeping telemetry data.

Acronyms

CNES Centre National d'Études Spatiales. 2, 4

GEO Geostationary Orbit. 1, 2

IMEI International Mobile Equipment Identity. 4, 5

LEO Low Earth Orbit. 1

MO Mobile Originated. 5

MOMSN Mobile Originated Message Sequence Number. 5

MT Mobile Terminated. 5

MTMSN Mobile Terminated Message Sequence Number. 5

NORAD North-American Aerospace Defense Command. 3

OBC On-Board Computer. 5, 6

SBD Short Burst Data. 5

SCC Simple Control Center. 4, 5

TC Telecommand. 5

TLE Two Line Element. 3

UART Universal Asynchronous Receiver Transmitter. 5, 6

References

[1] https://celestrak.com/norad/elements/iridium.txt.

[2] M. Masmano P. Arberet A. Crespo, I. Ripoll and J.J. Metge. Xtratum: An open source hypervisor for tsp embedded systems in aerospace.

[3] ISIS Innovative Solutions In Space B.V. Communication systems brochure. Motorenweg 23, 2623CR, Delft, The Netherlands, June 2006.

[4] I.A. Timbai I.V. Belokonov, A.V. Kramlikh. Low-orbital transformable nanosatellite: research of the dynamics and possibilities of navigational and communication problems solving for passive aerodynamic stabilization. In *Proceedings of 2th IAA Conference on Dynamics and Control of Space System*, Roma, Italy, March 2014.

[5] Jaanus Kalde. Uhf communication system for cubesatellite. In *Master's thesis*, University of Tartu, 2015.

[6] Robert Rathburn Margaret M. McMahon. Measuring latency in iridium satellite constellation data services. 2005.

Appendix

A. State machine of the driver's core layer

B. Phone constellations coverages

C. Visibility function of altitude and inclination

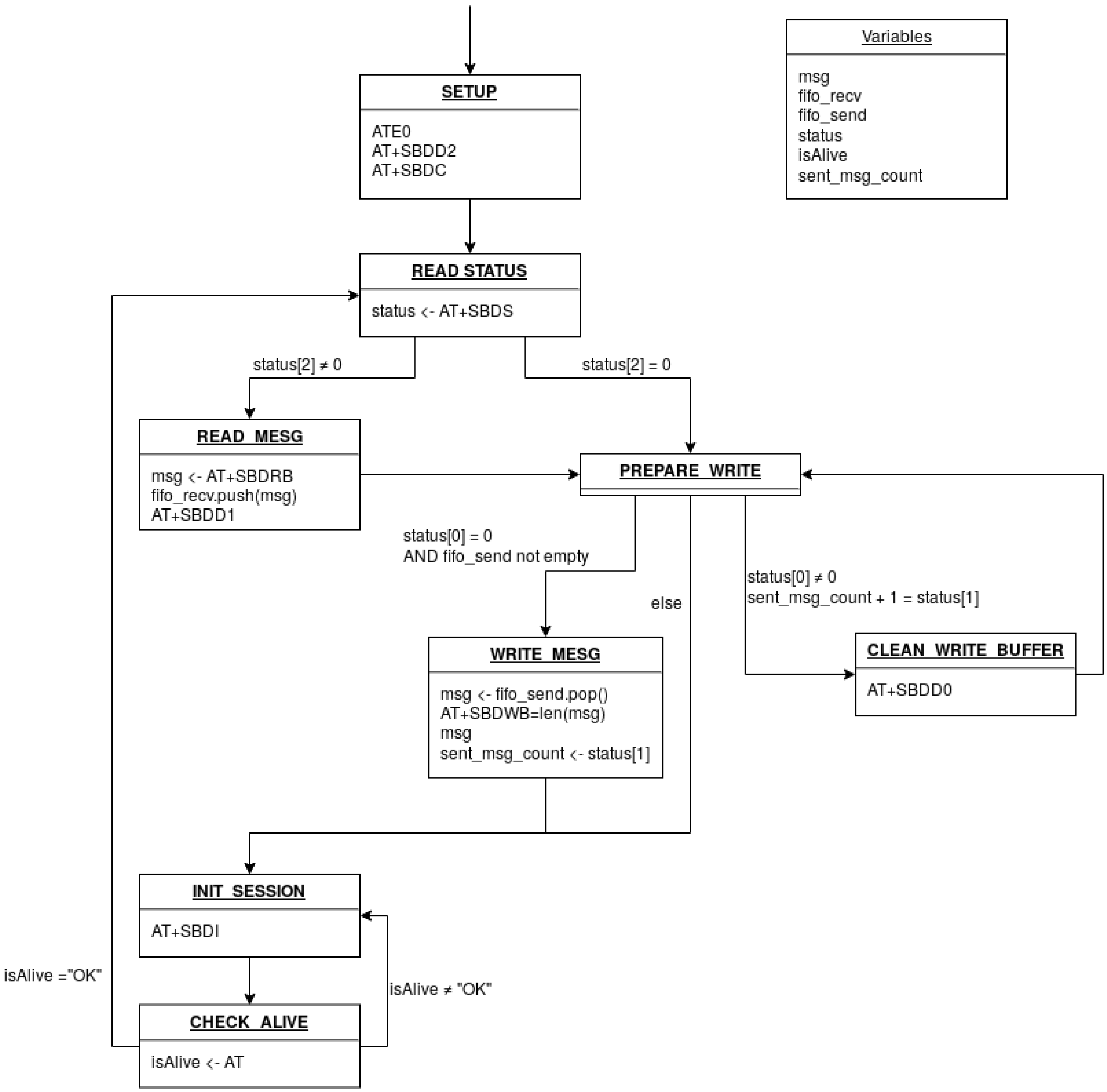
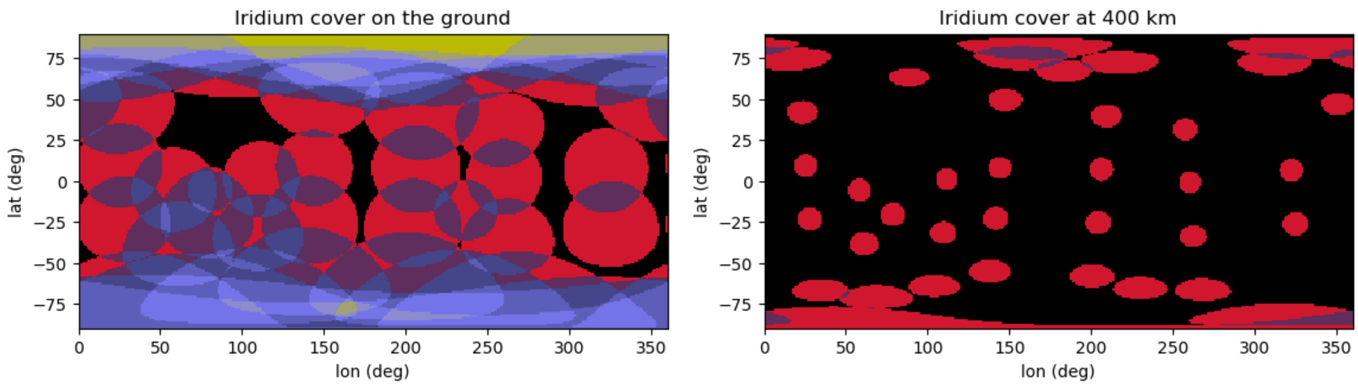


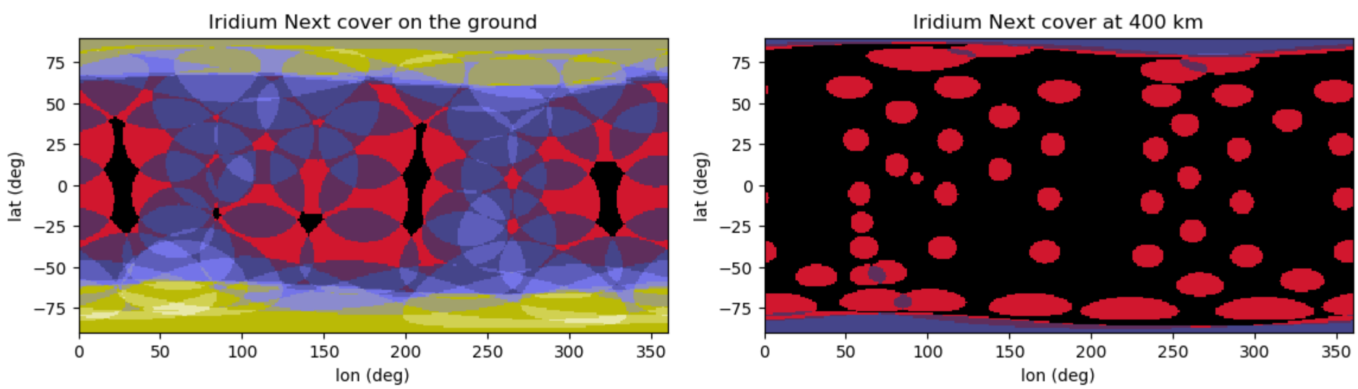
Fig. 7. State machine of the driver's core layer



(a)

(b)

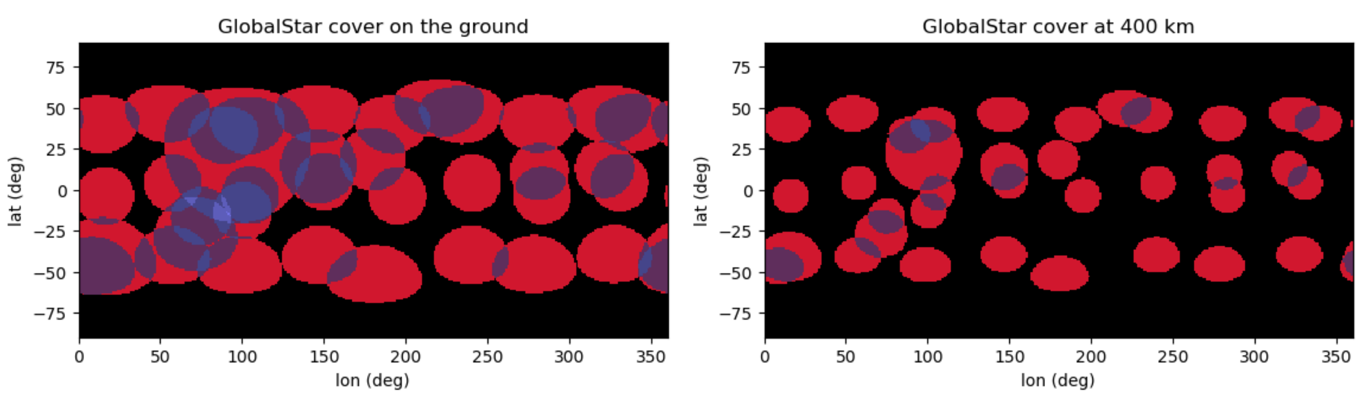
Fig. 8. Iridium



(a)

(b)

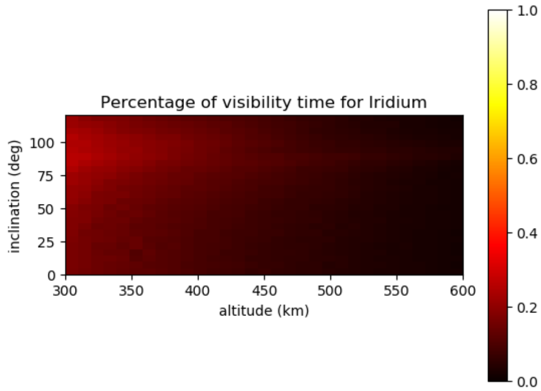
Fig. 9. Iridium Next



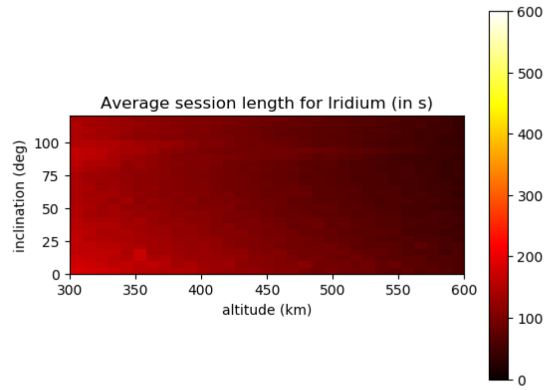
(a)

(b)

Fig. 10. GlobalStar

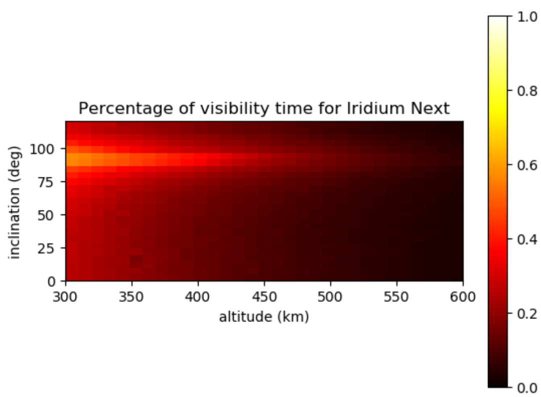


(a)

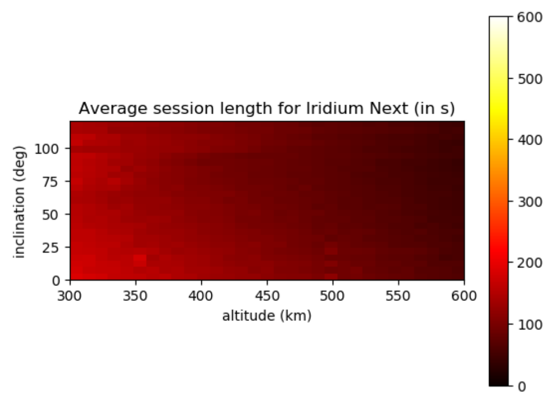


(b)

Fig. 11. Iridium

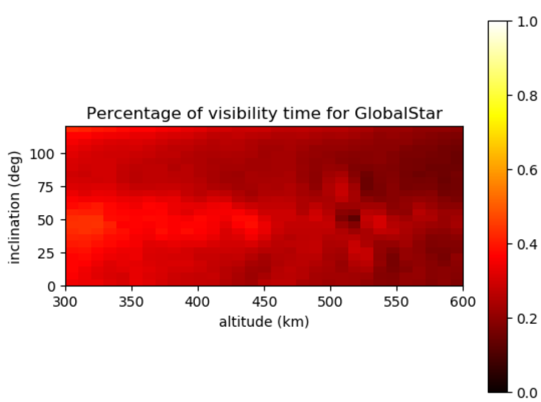


(a)

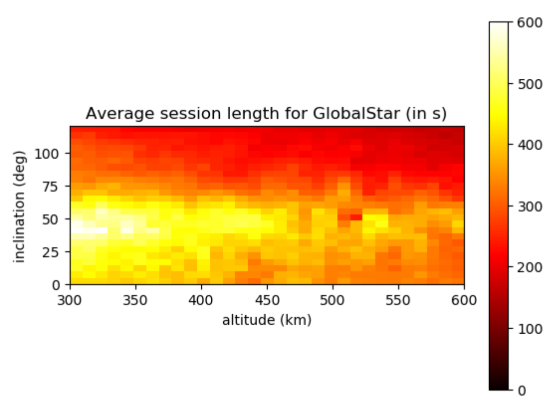


(b)

Fig. 12. Iridium Next



(a)



(b)

Fig. 13. GlobalStar